

embeo



smb filesharing development kit

Getting Started and Samples

Getting Started Guide
rev. 1.4

1 Introduction to SMB

The SMB protocol, CIFS, or “Windows File Sharing” as it is more commonly known, is the de facto file and printer sharing system in use today. An SMB server shares out file and printer resources over the network. SMB client software can access these resources or ‘shares’ and use them according to its purpose. The most recent protocol version has been a built-in component to all versions of Microsoft Windows since the release of Windows 95. There have also been many software suites written, the most popular being SAMBA, to expand the compatibility to any UNIX-like operating system (including Linux and Mac OS X). As a result, SMB can be found throughout corporate and home networks.

In SMB there are clients and servers. The server provides the resources such as files to clients. When a server provides or shares out a resource to the client, it is called a **share** (or a shared folder). A server can have many shares, so shares are given unique names. If, for example, a server shares out a folder named **Files** (possibly located in C:\Files), the share can be given the same name as the folder, or it can be named something else (such as **JohnsFiles**).

Each share can have its own password or a set of users with passwords that have access to the share. Shares can also have no password set.

SMB shares are accessed via the “Network Neighborhood” or “My Network Places” on Windows machines, or directly through the use of UNC paths, [\\computername\sharedfolder\resource](#) (or smb://computername/sharedfolder under a UNIX-like OS). On many networks, SMB shares are mapped to a drive letter such that the drive letter points to a remote folder. The following table shows an example of the relationship:

<u>Local Path</u>	<u>Server Shared Folder Path</u>	<u>UNC</u>
M:\	C:\Home\Bob\	\\server\Bob\

The network names of the clients and servers are resolved to IPs using the NetBIOS protocol. NetBIOS is a LAN protocol that uses broadcast, multicast, or directed UDP datagrams depending on the node type and configuration. Note: NetBIOS does not resolve beyond the local network.

2 Using the SMB Application Kit to Connect to a PC

In order to use the ThinShare SMB App kit software to access a shared drive/folder on a PC you must first setup the PC to accept this type of connection. For help on configuring the PC, refer to the Help Sections in this guide and Section 3.2 in the ThinShare manual.

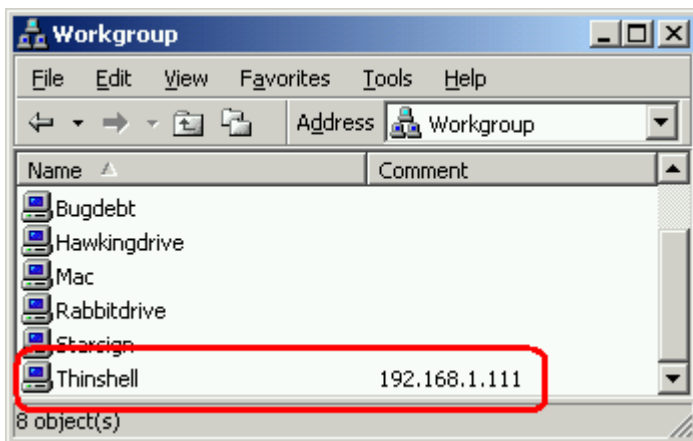
Set the following macros appropriately to use the sample programs:

```
#define SMB_NODE_NAME    "thinshell" //unique network name of this device
```

The `SMB_NODE_NAME` is used to enable auto registration of SMB node names (computer name). This will allow the Rabbit module to be accessed by this name over the network. For example, when a sample is running and periodically calling `smb_tick`, the command prompt can be used to ping the Rabbit.

```
C:\> ping thinshell
```

Also, the Rabbit will appear in "My Network Places" or "Network Neighborhood" with the name "thinshell":



Note: You must set the workgroup to a pre-existing workgroup containing other servers in order to see the Rabbit in the list. For this reason the workgroup named WORKGROUP was chosen as it is commonly used.

Next retrieve the computer name for the PC. Right click "My Computer"->Properties->Computer Name. (Refer to Section 3.2.1 of the ThinShare manual for help.)

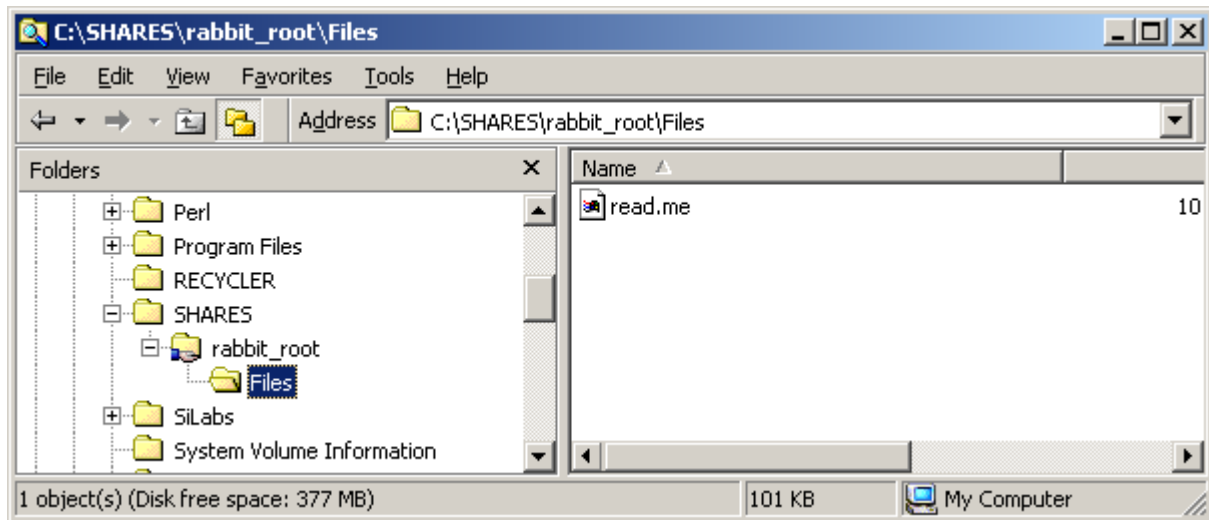
```
#define SMB_SERVER      "my_pc_name"      //remote server name goes here

//NETBIOS settings, see Thinshare manual for more details
#define NETBIOS_B_NODE          //Use NETBIOS Broadcast node to save memory
#define SMB_BROWSE_ENABLE      //creates icon in network neighborhood
```

In order to access the shared folder you can use the default "guest" account which usually has no password. You can also use your username/password or create a new user. For more information on creating users, use Windows help and search for "creating users".

```
#define SMB_USER        "guest"          //user name
#define SMB_PASSWD     ""                //and password
```

The `SMB_SHARE` macro is the name of the shared drive or folder. Note: If there are sub-folders within the shared folder, they can be accessed by using the full path starting with the shared drive/folder. For example, if we have the following:



In this example, the “rabbit_root” folder (`C:\SHARES\rabbit_root`) is shared. Every sub-folder and file is inherently shared. So the folder “Files” and its contents are shared, along with the file “read.me” which is inside of “Files”.

```
#define SMB_SHARE "rabbit_root" //shared root folder
//connect to the SMB_SERVER's SMB_SHARED folder
handle = smb_connect( SMB_SERVER, // "my_pc_name"
                    SMB_SHARE, // "rabbit_root"
                    SMB_USER, // "guest"
                    SMB_PASSWD ); // ""
```

Note: To access the file “read.me” you must refer to it by full path relative to the SHARE, so “Files\read.me” is valid but “C:\SHARES\rabbit_root\Files\read.me” is NOT valid.

```
smb_open(handle, "Files\\read.me", SMB_O_RDONLY); //open for read only
```

Note: All of the above macros do not have to be used; they are setup in the samples for ease of use (in `smb_config.lib`). You can make function calls and pass dynamic network settings as parameters. You can also have multiple connections because the `smb_connect` function returns a handle that is used to access the specified shared drive/folder.

3 Samples

Most all of the samples use the following macros which are defined in **smb_config.lib**:

```
REMOTE_SERVER // Must refer to a valid server name
REMOTE_SHARE  // Must refer to a valid share on the server
REMOTE_USER   // User must have access to the share on the server
REMOTE_PASSWD // User's password
```

The user generally needs read-write access to the share. Some samples have different or additional requirements.

3.1 hello_world_txt.c – “Hello World.txt”

“Hello World.txt” is the closest to a hello world program for SMB you can probably get. This sample creates a new file named “Hello World.txt” on the server’s SMB share and writes “Hello World!” into it.

Requirements: The user has create and write-access to the share.

3.2 smbshell.c – “ThinShell”

ThinShell provides an interactive command-line shell with a large number of available commands. Many common commands are available and virtually every aspect of the ThinShare SMB Library is utilized.

To get help in the shell type "help" or "?". To get help for a specific command type "help commandname". Some examples of how to use ThinShell:

```
connect thindrive samples rabbit rabbit
dir
cat readme.txt
```

Requirements: Varies per command.

3.3 cat.c – “View File”

cat is named after the UNIX program of the same name. It is like the UNIX cat, or the DOS program type. It simply prints out the contents of a text file to the screen, in this case the Dynamic C stdio window.

cat will by default look for the file \\REMOTE_SERVER\REMOTE_SHARE\readme.txt, which will usually be \\thindrive\samples\readme.txt

Requirements: The input file exists on the server and the user account has read-access to the file.

3.4 cp.c – “Copy File”

cp is named after the UNIX program of the same name. It is like the UNIX cp or the DOS program copy. Two files are specified, the source and destination. The source file is copied into a new file named destination.

cp will by default copy “readme.txt” to “Copy of readme.txt”.

Requirements: The source file exists on the server and the user account has read-write access to the share.

3.5 comprehensive.c – “Comprehensive File I/O”

Comprehensive is a fairly comprehensive sample covering almost all of the filesystem based functions. This sample generates its own files and is thus self-sufficient. It starts by creating a directory and a file in that directory. The file is then the subject of the rest of the sample: it is written to, read from, renamed, etc.

Requirements : The user account must have create and read-write access to the share.

3.6 network_smb_dump.c – “Server/Share Information Dump and Printout”

The Network SMB Dump sample generates a list of machines in the device’s current workgroup. It then gets the list of shares on the server specified. This list can be written to the server as either a text file or as a file to be printed (a printer spool file). If the print option is attempted, the printer must support line-printer style print access (non-postscript printers such as many inkjet or dot-matrix printers work well).

Requirements: The workgroup must have a master browser that is accessible, the user account must have access to list shares on the server, and the user account must have access to create a print spool file on the printer (if printing mode is enabled).

3.7 Remote Control (remote_control.c)

The remote control sample program executes script files off of specific script folder on the remote SMB server. The idea is a user copies a script into the folder to do a specific task or list of tasks. The rabbit parses the script and executes commands within it. After the script is finished executing, the rabbit deletes the script file. Output is appended to a file called SCRIPTNAME.log. The commands can be user-specified. The script files support variables and comments as well. Script files are named by default *.ts (thinshare script).

Remote Control program flow

1. Connect to remote server ([\\REMOTE_SERVER\REMOTE_SHARE](#))
2. Open scripts directory (REMOTE_DIR in REMOTE_SHARE on the server)
3. Wait for a specified interval (30 seconds)
4. Scan scripts dir for *.ts files
5. For each *.ts file found (SCRIPTNAME.ts)
 - a. Parse line from script
 - b. Execute command or skip to next line
 - c. Write output for line to output file (SCRIPTNAME.log)
 - d. Repeat (a thru c) until EOF
 - e. Delete input script file
6. Repeat (Goto #2)

There are several sample scripts located in the ThinShare RemoteScripts directory in the installation folder. Here is a simple example script file:

```
echo "This is the Remote Script Echo Sample"

set SOMEVARIABLE hello
set another_var "echo and variable"

# Put quotes around variable names you don't want to be replaced...
echo "The variable $SOMEVARIABLE =" $SOMEVARIABLE
echo this is another $another_var example
usercommand parameter1 parameter2
echo done!
```

User-defined script keywords can be added by adding a command to the RC_COMMANDS array.

```
MAKECOMMAND(usercommand, 2, 2, "PARAM1 PARAM2 -- User-defined command")
```

Just put a copy of the script file to be executed in REMOTE_DIR on the REMOTE_SERVER that the remote_control.c sample connects to and run the sample program. Within ~30 seconds the script should execute on the rabbit!

3.8 Remote Logging Facility (remote_log.c)

The remote logger is a handy library for creating log files over an SMB connection. This sample demonstrates the functionality of the library (remote_log.lib).

The logging service writes log entries to a log file on an SMB server. It writes a single-line log entry in the log file each time rl_log() is called. This log entry includes the current date and time and the text passed to rl_log(). Also, the logger timestamps when it was started and stopped ("Log Started", "Log Closed"). Logs are appended to the file.

This is how the log files will look after running the remote_log sample:

Contents of log.txt

```
Tue Aug 22 22:17:26 2006 Log Started
Tue Aug 22 22:17:26 2006 This is message one in log.txt
Tue Aug 22 22:17:26 2006 This is message two in log.txt
Tue Aug 22 22:17:26 2006 Switching to log2.txt now
Tue Aug 22 22:17:26 2006 Log Closed
```

Contents of log2.txt

```
Tue Aug 22 22:17:26 2006 Log Started
Tue Aug 22 22:17:26 2006 This is message one in log2.txt
Tue Aug 22 22:17:26 2006 This is message two in log2.txt
Tue Aug 22 22:17:26 2006 Thats all for the logging sample
Tue Aug 22 22:17:26 2006 Log Shutdown
```


4 Using the Network Attached Storage (NAS) Drive

The SMB Application kit includes an Eagle Consus T-Series NAS drive. This hard drive is accessible to the Rabbit and PCs via the SMB protocol. It is intended to be used with the sample programs included with the App kit. Many of the sample programs require an SMB server, and either the NAS drive or a shared folder on a PC can serve this purpose.

By default the node name of the NAS is set to “**STORAGE-XXXX**”. The ‘XXXX’ represent the last four digits of the drive’s MAC address which can be found next to the RJ-45 Ethernet Port. The default details are summarized below:

Default Node Name:	STORAGE-XXXX (XXXX = last four digits of the MAC address)
Default User:	admin
Default Password:	admin
Default Share:	Public

In order to use the NAS drive with the sample programs, it must be configured correctly for your network. It can be configured via its built-in web server to match the values in `smb_config.lib` (**recommended**), or `smb_config.lib` may be modified to match the NAS drive’s settings. If you are not going to use a DHCP server make sure to give the NAS an IP address on the same network as the Rabbit module. Refer to the Eagle NAS drive manual for more information about configuring the drive’s network settings.

To configure the NAS device via the built-in web server, point your web browser to <http://storage-XXXX/> (where XXXX = last four digits of MAC address. Or use the shortcut <http://storage/>) and enter the default username/password of admin/admin. First, change the NAS host name to the value of **REMOTE_SERVER** in `smb_config.lib` (defaults to `thindrive`). Create a share that matches the value of **REMOTE_SHARE**. If creating a share with authentication, remember to fill in **REMOTE_USER** and **REMOTE_PASSWD** with the correct values in order for the samples to work correctly. (Note: once the NAS host name is changed, the web configuration page is accessed via the new name. i.e. <http://thindrive/>).

Once the drive is configured, compile and run the `smbshell.c` sample on the Rabbit. Once the shell is running, type the following in the STDIO window of Dynamic C:

```
rabbit@:\> ping thindrive
received ping from 192.168.1.139: icmp_seq=0 time=0ms
received ping from 192.168.1.139: icmp_seq=0 time=0ms
received ping from 192.168.1.139: icmp_seq=0 time=0ms
received ping from 192.168.1.139: icmp_seq=0 time=0ms
```

Note: If you do not receive a ping back from the NAS drive at the configured name (in this case, `thindrive`) then there is a problem with your network configuration. You may want to connect to the NAS drive’s web page and find its IP, then try pinging the IP from the Rabbit or a PC.

Now connect to the NAS drive’s shared folder:

```
rabbit@:\> connect thindrive samples
connecting to @: \\thindrive\samples...
connected
```

At this point we're connected to the "Samples" share on the NAS, and we can run commands like `dir`, `cd`, `md`, `del`, etc (type "help COMMAND" for help on a specific command).

```
@thindrive:\Samples\> dir

volume in drive is Samples

Directory of \

02/15/2007  09:31a      <DIR>          .
02/15/2007  09:31a      <DIR>          ..
              0 File(s)          0 bytes
              2 Dir(s)

@thindrive:\Samples\> mkdir NewFolder
creating directory NewFolder...
created directory
@thindrive:\Samples\> write NewFile.txt
This is a new file!

19 bytes written to file NewFile.txt.
@thindrive:\Samples\> type NewFile.txt
This is a new file!
@thindrive:\Samples\>
```

After successfully creating a new folder and file, we've verified the NAS drive is fully accessible by the Rabbit. Look over the sample programs and the ThinShare manual for help on writing your own SMB-connected programs using ThinShare!

5 Appendix: Windows Help for Sharing Folders/Drives

5.1 Help Sharing a Folder/Drive (From Windows 2000 Help)

To share a folder or drive with other people

1. Open Windows Explorer, and then locate the folder or drive you want to share.
2. Right-click the folder or drive, and then click **Sharing**.
3. On the **Sharing** tab, click **Share this folder**.
4. To change the name of the shared folder or drive, type a new name in **Share name**. The new name is what users will see when they connect to this shared folder or drive. The actual name of the folder or drive does not change.
5. To add a comment about the shared folder or drive, type the text in **Comment**.
6. To limit the number of users who can connect to the shared folder or drive at one time, under **User limit**, click **Allow**, then enter a number of users.
7. To set shared folder permissions on the shared folder or drive, click **Permissions**.
8. To set up this shared folder to be used offline, click **Caching**.

Notes

- To share folders and drives, you must be logged on as a member of the Administrators, Server Operators, or Power Users group.
- To open Windows Explorer, click **Start**, point to **Programs**, point to **Accessories**, and then click **Windows Explorer**.
- If the **Sharing** tab is not visible, try using the **Services** snap-in to start your Server service.
- If the folder is already shared, click **New Share** and then type the new share name.
- You can hide the shared folder from browsing by typing \$ as the last character of the share name. Users will not be able to see this shared folder when they browse using My Computer or Windows Explorer, but they can map to it.
- In Windows 2000 Professional, the maximum user limit is 10 regardless of the number you type in **Allow**.
- You can use the Shared Folders snap-in to create and manage shared folders, view a list of all users who are connected to the shared folder over a network and disconnect one or all of them, and view a list of files opened by remote users and close one or all open files.

5.2 Help Sharing a Folder/Drive (From Windows XP Help)

To share a drive or folder on the network

1. Open Windows Explorer, and then locate the drive or folder you want to share.
2. Right-click the drive or folder, and then click **Sharing and Security**.
 - If you are sharing a drive, on the **Sharing** tab, click **If you understand the risk but still want to share the root of the drive, click here**.
 - If you are sharing a folder, go to the next step.
3. Do one of the following:
 - If the **Share this folder on the network** check box is available, select the check box.
 - If the **Share this folder on the network** check box is not available, this computer is not on a network. If you would like to set up a home or small office network, click the **Network Setup Wizard** link and follow the instructions to turn on file sharing. Once file sharing is enabled, begin this procedure again.

Notes

- To open Windows Explorer, click **Start**, point to **Programs**, point to **Accessories**, and then click **Windows Explorer**.
- When you share a drive or folder, anyone who has user access to the network can read the contents of that drive or folder. If you want to remove the drive or folder from the network, follow steps 1 and 2 above to get to the **Sharing** tab. On the **Sharing** tab, in **Network sharing and security**, clear the **Share this folder on the network** check box.
- To allow complete access to the contents of your shared drive or folder, follow steps 1 and 2 above to get to the **Sharing** tab. On the **Sharing** tab, in **Network sharing and security**, select the **Allow network users to change my files** check box.
- To change the name of your folder on the network, in the **Share name** text box, type a new name for your folder. This will not change the name of the folder on your computer.
- If you are logged on as a guest (a guest account provides access to the computer for any user who does not have a user account on the computer), you cannot create a shared folder.
- The **Sharing** option is not available for the Documents and Settings, Program Files, and WINDOWS system folders. In addition, you cannot share folders in other user's profiles.